Kubernetes集群的部署方式及详细步骤

前言：

相比二进制在主从节点上部署K8S集群，部署方式非常的繁琐，出于安全方面的考虑，各组件之间通信都需要配置CA和证书。缺点就是如果master宕机了，需要手动进行启动这些系统级的守护进程

所以，此次采用kubeadm工具进行部署，由k8s官方所提供的专门部署集群的管理工具。

每一个节点主机上包括master节点都要手动安装并运行docker，同时也都要手动安装并运行kubelet。如果将第一个节点初始化为master节点，在执行初始化这个步骤，其实就是通过kubeadm工具将API Server、etcd、controller-manager、scheduler各组件运行为Pod，也就是跑在docker上。而其他node节点，因已经运行了kubelet、docker组件，剩下的kube-proxy组件也是要运行在Pod上

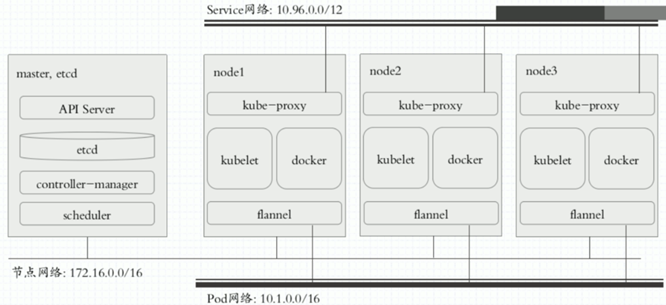
kubelet：负责能运行Pod化容器的核心组件

docker：运行容器的引擎

但是以上master节点上的Pod都是静态Pod（static Pod），并不受k8s自身管理，只是运行为Pod形式而已，也可运行为自托管Pod

K8S docker 在部署集群上，存在版本的兼容行问题，请根据实际情况选择兼容性版本安装。

架构

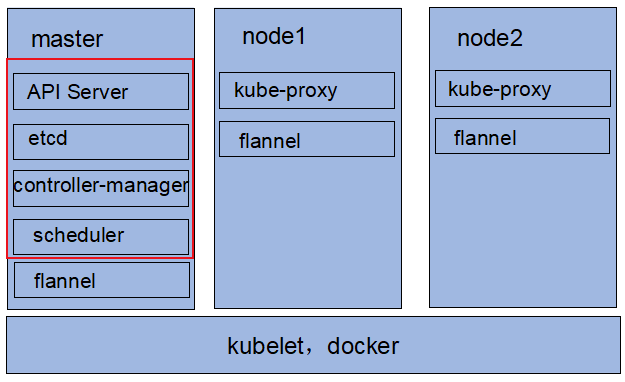


此次部署中：

节点网络135.251.206.39(master), 135.251.206.38, 135.251.206.37

Pod网络：10.244.0.0/16

Service网络：10.1.0.0/16



flannel组件也是运行在Pod上的系统级守护进程，起到各node节点通信作用

部署流程

1、master、nodes节点要先安装kubelet、kubeadm和docker

2、在master节点上运行kubeadm init进程进行初始化集群

3、将各node节点加入集群中：kubeadm join

部署操作(以下操作在使用节点均要执行)

1. 确保集群中的所有节点，服务器时间是同步的。

可参考以下文档进行部署

<https://www.cnblogs.com/huhyoung/p/9657253.html>

1. 服务器可配置hosts文件进行域名解析
2. 关闭防火墙、selinux和swap

systemctl stop firewalld

systemctl disable firewalld

setenforce 0

sed -i "s/^SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config

swapoff -a

sed -i 's/.\*swap.\*/#&/' /etc/fstab

4.第一次安装打开路由转发功能

echo "1" > /proc/sys/net/ipv4/ip\_forward

5.配置内核参数，将桥接的IPv4流量传递到iptables的链

cat > /etc/sysctl.d/k8s.conf <<EOF

net.bridge.bridge-nf-call-ip6tables = 1

net.bridge.bridge-nf-call-iptables = 1

EOF

执行 sysctl --system ,让配置生效

6. 配置国内Kubernetes源

cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=0

repo\_gpgcheck=0

gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg <https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg>

7. 配置 docker 源

wget [https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo -O /etc/yum.repos.d/docker-ce.repo](https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo%20-O%20/etc/yum.repos.d/docker-ce.repo)

8. 安装docker、kubeadm、kubectl、kubelet

yum list docker-ce --showduplicates | sort -r

yum install -y docker-ce-18.06.1.ce-3.el7

systemctl enable docker && systemctl start docker

yum install -y kubelet kubeadm kubectl

systemctl enable kubelet

在Master节点执行

进行Kubernetes集群初始化(重置初始化命令: kubeadm reset)

kubeadm init --kubernetes-version=1.15.3

--apiserver-advertise-address=192.168.3.70

--image-repository registry.aliyuncs.com/google\_containers

--service-cidr=10.1.0.0/16

--pod-network-cidr=10.244.0.0/16

红色，其中主机地址根据实际情况变更

在初始化过程中出现如下error，处理方案



1：vim /etc/sysconfig/kubelet #加入以下参数

KUBELET\_EXTRA\_ARGS="--fail-swap-on=false"

2：初始化命令中加入参数 -ignore-preflight-errors=Swap

最后会生成token如下

Kubeadm join 135.251.206.37:6443 --token kekvgu.nw1n76h84f4camj6 \

--discovery-token-ca-cert-hash sha256:4ee74205227c78ca62f2d641635afa4d50e6634acfaa8291f28582c7e3b0e30e

配置kubectl工具

系统由admin.conf文件启动kubernetes,如果不配置会起不来

cp /etc/kubernetes/admin.conf /root/.kube/config

部署flannel网络

kubectl apply -f <https://raw.githubusercontent.com/coreos/flannel/a70459be0084506e4ec919aa1c114638878db11b/Documentation/kube-flannel.yml>

在Node 节点执行

master上生成的token那段直接在各node执行

之前保证node的kubelet 未启动

Kubeadm join 135.251.206.37:6443 --token kekvgu.nw1n76h84f4camj6 \

--discovery-token-ca-cert-hash sha256:4ee74205227c78ca62f2d641635afa4d50e6634acfaa8291f28582c7e3b0e30e

最后，在master主机上获取nodes的状态，成功即部署完成。

Kubectl get nodes

kubuectl get cs

kubectl get pod -n kube-system